EXHIBIT K

Journal of Systems Management

October 1993

SECTION: Vol. 44, No. 10 Pg. 10-15+; ISSN: 0022-4839; CODEN: JSYMA9

LENGTH: 4685 words

HEADLINE: On selecting appropriate technology for knowledge systems

BYLINE: Slater, John R; Hazen, Susan J; Sakthivel, Sachi

BODY:

Traditionally, computers have been used to solve structured problems for which solution steps are defined explicity. The computer programs used to solve such problems behave in a predictable manner and can only handle foreseen problems. In contrast, human beings solve problems in ways that cannot be defined explicitly. For example, the process a physician follows to diagnose a patient's illness or the process a manager follows to decide on a particular investment cannot be defined explicitly. A branch of computer science called Artificial Intelligence (AI) has the chief objective of finding out if we can use computers effectively in such problem solving situations. AI deals with computer-based systems that exhibit human characteristics such as understanding, reasoning and decision making. Artificial Intelligence has evolved into several disciplines, notably Human Perception (speech understanding, vision understanding, natural language processing, etc.), Robotics and Knowledge Systems. Whereas the first two disciplines are primarily laboratorybased prototypes, knowledge systems have active applications in business.

Knowledge systems extend the use of computers beyond their traditional area of data processing and into that of intelligent reasoning. Traditional systems supply information to enable decision making at all levels of management. They capture data from within and outside of the organization, use algorithmic procedures to manipulate them and provide information. Knowledge systems make decisions involving intelligent reasoning in narrowly defined fields such as machine maintenance, loan processing and forecasting. They capture expensive and rare corporate expertise about solving a problem, use procedures such as inference mechanisms to manipulate that knowledge and make the decision. A set of characteristics such as whether data are qualitative or quantitative, how these data are represented and how they are manipulated can be used as guidelines to choose between knowledge-based and traditional systems design.

Knowledge-based computer systems put the expertise in the hands of non-experts for making consistent decisions throughout the organization. This improves the corporate decision making efficiency, and experts are now free to concentrate on more difficult problems. Knowledge systems are useful in training new experts. Knowledge systems provide opportunities for organizations to create competitive advantages by employing computers in knowledge intensive areas which have not been previously exploited.

There are two classes of knowledge systems depending on the technology they are using: Expert Systems (ES) and Artificial Neural Network (ANN) Systems. In Expert Systems, knowledge about a specific problem is acquired from human experts and stored in a database called a knowledge base. By processing this knowledge base, ESs mimic the decision making process of human experts. Artificial Neural Networks offer an alternative approach to solving knowledge intensive problems by simulating approximate models of the biological neural network found in human brains. Although both ESs and ANNs are used in solving knowledge intensive problems, there are many differences in their construction and application. This article explains the essential features of ESs and ANNs, discusses the advantages and issues in using each of these techniques and indicates how each is well suited to solve specific kinds of knowledge intensive problems. This article also presents a case to illustrate an application suitable for both ESs and ANNs.

EXPERT SYSTEMS

Expert Systems have three main parts: the knowledge base, the inference engine and the explanation system. In addition, they may have a user interface and a knowledge acquisition system as shown in Figure 1. (Figure 1 omitted)

The knowledge base consists of facts about the problem. It also consists of rules which are the mechanisms to infer new facts. The inference engine contains the mechanism to search the knowledge base and to provide conclusions based on these searches. The explanation system provides a trail of reasons necessary to show how a conclusion was reached. The inference engine and the explanation system may have user interfaces to facilitate convenient user interaction with the ES. The knowledge acquisition system includes input/output interfaces to acquire, store and maintain knowledge. ESs normally provide deterministic results. However, by associating certainty levels with facts and rules, the certainty of a decision can be determined. Certain ESs can also handle fuzzy logic that involves subjective ideas. There are four main methods to the design of ESs: production systems, frame-based systems, semantic networks and logic programming. The differences among these methods arise out of their knowledge representation schemes and inference mechanisms.

PRODUCTION SYSTEMS

In production systems, knowledge is represented using IF....THEN rules, and hence, they are also called rule-based systems. When facts are presented, the production system makes a pass through the rules and selects rules that match the facts. One of these selected rules will be executed to produce new facts which in turn will be matched with other rules. For example, consider the following rules in a simplified production system for car maintenance:

IF gas consumption is high & exhaust fumes smell of gas

THEN problem is spark plugs

(more facts and rules)

When facts 'gas consumption is high' and 'exhaust fumes smell of gas' are presented to this production system, it will select the IF rule, execute it and infer that the problem is with the spark plugs. This new fact will be matched with other rules and if a match is found, then it will be selected to infer more facts. The selection, matching and inference will continue until a match no longer occurs at which time the last inferred fact would be the conclusion. Production systems are easy to understand and construct. Since the system requires a set of rules and facts, they are only useful to solve structured problems.

FRAME-BASED SYSTEMS

In frame-based systems, knowledge is represented hierarchically in terms of frames and slots. A frame represents an entity or one's expectations in stereotypical situations. A frame consists of a set of slots. The slots represent the typical attributes of the frame. Each slot has certain values associated with the slot. The value of a slot can also be the name of a frame to establish hierarchical relationship between the two frames. For example, consider the following frame-based system:

In this system, transportation, a way of moving people, is represented by a frame. The three slots under this frame have values 'plane', 'train', and 'automobile' which are frames, and thereby forming hierarchical relationships with the frame 'transportation'. The frame 'automobile' has the slots, Ford Escort, Toyota Camry, and Porsche. Each of these slots has values about the number of passengers per car and the cost of driving the car per mile. When presented with the question,'What is the most economical method of transporting two persons for 100 miles?', the inference engine will search the knowledge base (exhaustively or using a heuristic) and may determine Ford Escort as the answer.

Frame-based systems are used to represent wellunderstood and stereotyped situations. An advantage of this method is its standard semantics. A disadvantage of this method is that for large applications, the knowledge base tends to be large and searching such knowledge bases may be time consuming. Therefore, the efficiency of this approach depends on its search strategy.

SEMANTIC NETWORKS

In semantic networks, knowledge is represented in terms of objects (or concepts) and relationships (or links) among them. Whereas frame-based systems are organized hierarchically, semantic networks can be complex. For example, consider the following semantic network:

In this semantic network, 'Joe is a Person', 'Jane is-a Person', and 'Joe likes Jane' are represented using 'Joe', 'Jane', and 'Person' as objects, and 'is-a' and 'likes' as relationships among them. These relations are associated with an infer-

ence mechanism which allows the inheritance of properties downward along the relations: if a Person has a job, then Joe and Jane have a job. The inference is done by starting with the object (Person), navigating down along the relations (is-a), and associating the property of 'Person has a job' to other objects (Joe and Jane).

The use of hierarchical relationships in frame-based systems places a limitation in describing complex relationships among objects which is overcome in semantic networks. However, this is also a problem because their semantics are not standard.

LOGIC PROGRAMMING

In these Expert Systems, facts and rules are represented in the logic of predicate calculus. Using a process called unification, the inference engine can search facts and rules to infer new facts. For example, facts such as 'Joe is taller than Jane' and 'Jane is taller than John' can be represented as:

taller (joe,jane).

taller (jane,john).

A rule now can be specified to infer new facts. For example, the rule

taller (A,C):-taller(A,B),taller(B,C),

specifies that if A is taller than B and B is taller than C, then A is taller than C. In logic programming, A, B and C are called variables and are denoted in upper case letters. This rule, when applied to the facts of Joe, Jane and John, will infer that Joe is taller than John. The advantage of this approach is that the semantics are well understood.

ARTIFICIAL NEURAL NETWORK SYSTEMS

ANNs simulate the massively parallel, interconnected network of biological neurons in the human brain. Although the intricacies of human brain functions and the complex cognitive mechanism used by the brain are not fully understood, the progress of research has enabled the design of ANNs with remarkable results. In this section, we will present a simplified description of the human brain and its function as well as how an ANN resembles a brain. For a detailed description, please see references 6,7.

The human brain consists of more than 100 billion interconnected elements called neurons. The number of interconnections among these neurons may exceed 100 trillion. A neuron consists of the cell body, dendrites and axons, as shown in Figure 2. (Figure 2 omitted)

Dendrites acting as receptors of signals pass them to the cell body through junctions called synapses. Synapses determine the strength of signals and pass them to the neuron. Since a neuron may be interconnected to many other neurons, several signals received at the cell body will be summed. If the level of such summed signals exceeds a 'threshold limit', then the cell is excited. An excited cell fires and sends its signal down its axon to surrounding neurons. Neurons are continually receiving, processing and transmitting signals but a single neuron contains no intelligence per se. Instead, the intelligence in the human brain is inherent in its interconnected structure of neurons.

The structure of an Artificial Neural Network resembles that of a human brain. The processing elements in an ANN, analogous to neurons in the human brain, have inputs, weights, a summation block, an activation function and outputs (see Fig. 3). (Fig. 3 omitted)

Inputs I1, I2, etc. may come from other processing elements or as original signals entering the network. Weights, analogous to synapses, determine how much influence the signal may have on the processing element. The summation block, analogous to cell body, mathematically combines these weighted inputs and presents it to the activation function. The activation function depending on its type (there are several types) and the level of the summed signals determines the output. Intelligence in an ANN is stored in the pattern of interconnections and the weight or strength of the connection between two elements.

ANNs have the ability to learn by training. Learning changes the interconnections among neurons, and hence, the intelligence of the ANN. The learning law can be one of supervised, reinforced and unsupervised methods. In supervised learning, the ANN receives sample inputs and outputs of the problem. For each input, the network will adjust its connection weights until it can produce the given output. The connection weights are adjusted for the entire sample until the produced outputs are within an acceptable range of given outputs. The sample size of inputs anc outputs depends on the nature of the correlatior between the inputs and outputs. In reinforced learning, the network is not given the output

but told if it i valid or invalid. In unsupervised learning, only the inputs are provided; the network classifies the input with similar characteristics and produces the outputs. This process may take a long time; times suc as 24 hours are common in the learning process. Neural networks implemented in hardware have significantly shortened the learning time.

In ANNs, processing elements with the same summation block, activation function and learning law are grouped into structures called layers. An ANN consists of an input layer, a hidden layer, and an output layer, as shown in Figure 4. (Figure 4 omitted)

The output of a processing element in a layer can flow to any other element including:

* an element in the preceding layer

* an element in the follwing layer

* an element in the same layer

* itself

* an external output

Processing elements, layers of processing elements and network layers integrate to form an ANN.

ANNs are well suited for problems involving pattern recognition. This ability allows ANNs to serve as eyes for Robots, Character Recognition Devices, etc. In business, they are suitable for data analysis, especially to predict the value of a dependent variable based on the values of related independent variables. In such analyses, no assumptions need to be made regarding their statistical properties. These features make ANNs attractive for analysis of data and they may perform better than sophisticated statistical techniques.

ISSUES IN USING ESS FOR KNOWLEDGE SYSTEMS

The availability of ES shells with good user interfaces and development tools have simplified ES development. Users with some training can develop an ES using these shells. In addition to the advantages of being a knowledge system, ESs have the merit of having an explanation system. The explanation system's ability to explain how and why it reached a particular decision can inspire confidence among its users. ESs are useful when a series of qualitative decisions are made in the form of a decision tree and when they require user interaction.

However, ESs are only useful in a narrowly defined domain such as solving maintenance problems for a group of known machine models. Within such areas ESs work well but fail when a problem requires generalization or extrapolation. For example, a machine maintenance ES may fail if a new problem occurs or when it is extended to another model of the machine.

For ESs to work well, rules and facts must be clearly identified. Non-procedural problems for which we can identify the solution intuitively but cannot identify the rules clearly cannot be represented in the knowledge base. For example, consider a bank that needs to evaluate loans based on the applicant's income, credit history, outstanding loans and number of dependents. If the credit evaluator does the job intuitively and is not able to clearly specify the rules, an ES for loan evaluation cannot be designed. Such problems may be solved using a method called induction. In this process, a large number of example inputs and outputs are presented to the ES which will try to understand the relationship between them, and then use that relationship in problem solving. This method does not work well and has only limited applications. ESs are weak in areas requiring such data analysis.

To develop ESs, problems must be at least partially structured, and some experts must be available to articulate the facts and rules to create the knowledge base. Therefore, in developing ESs, knowledge acquisition becomes an important step. The success of this step could decide the success or failure of the ES. The nature and intricacies of the knowledge acquisition process requires a trained knowledge engineer with skills in interviewing, problem solving and structured documentation. Although ES shells are available, ESs are rarely developed by users.

Expert Systems do not perform as well as the human experts who helped create them. In fact, the performance of the human expert places a theoretical performance limit for the ES. Any improvement in ES technology can only help to narrow the gap between the ES and the human expert performance.

ISSUES IN USING ANNS FOR KNOWLEDGE SYSTEMS

Like ESs, Artificial Neural Networks are also useful only in narrowly defined problem domains involving pattern recognition and data analysis. They are useful in applications such as credit appraisal, bond rating, automated quality control and market forecasting. A major advantage of using ANNs is that one does not have to identify the rules. This feature is useful when the decision maker cannot clearly articulate the decision making process or when the decision making process is intuitive using certain inputs. ANNs excel in this area where ESs fail. Because one does not have to identify rules, the knowledge acquisition process is not difficult. However, the inputs and outputs must be correlated, and the developer needs to identify those inputs (independent variables affecting the decision) and the outputs (the decision). The development of ANNs does not depend on the skills of the developer or the knowledge engineer and puts the tool right in the hands of the expert. The availability of neural network shells simplifies such development.

Artificial Neural Networks have the exceptional ability to learn, and therefore, as they are used for a longer time they become more accurate in predicting. ANNs can easily adapt themselves when the environmental conditions of the problem or the relationship between inputs and outputs change.

ANNs can handle fuzzy inputs better than ESs. When an ANN is presented with incomplete or even contradictory data, it can choose the best match for the input. They can also handle noisy input. This feature makes it attractive for recognizing handwritten letters because most people do not write letters in the same way.

Since ANNs store knowledge in numerous processing elements and their interconnections, they are highly fault tolerant. Accidental destruction of a few processing elements or their interconnections may not affect the system.

ANNs may outperform human experts in certain areas such as forecasting. A statistical expert armed with sophisticated statistical modeling tools and computers may not be able to forecast better than an ANN. Since ANNs are not modeled after human experts, they don't import their inefficiencies. In addition, ANNs use a different model for forecasting using inputs, outputs and their correlations.

One disadvantage of using an ANN is that the system acts like a black box providing little insight into how the decision is made. Unlike ESs, they do not have an explanation system. Although the system may be developed carefully and validated for correctness, users may not have confidence in the decisions made. Although ANNs have been shown to provide excellent results comparable to statistical techniques such as regression and time series for forecasting, statistical experts have viewed their use with skepticism. In this respect, ANNs are viewed like traditional computer systems were in the 1950s.

SUMMARY OF ISSUES IN USING ESS AND ANNS

Both ESs and ANNs are useful for developing knowledge systems but each has certain strengths and weaknesses as summarized in Table 1. (Table 1 omitted) The table does not mean that ESs and ANNs are always useful only in mutually exclusive situations. For certain types of problems, such as when rules are available and there is a correlation between inputs and outputs, both may be useful. In such situations, ESs and ANNs can complement each other; while ANNs provide the output, ESs can provide an explanation why such output was produced. In any systems development process, validation of the system--by checking its correctness in system behavior and in providing the right output--is necessary. An ANN does not take as much time to develop as an ES, and can then be used to validate the ES development.

APPLICATION EXAMPLE--THE LOAN EVALUATION SYSTEM

The objective of the Loan Evaluation System (LES) is to evaluate mortgage loan applications in a commercial bank based on the customer's mortgage amount, total income, number of vehicles owned or on lien, amount in checking and savings accounts, outstanding loans and their payment amounts, employment history, credit history, the number of dependents, etc. The bank has processed many loan applications; their data and results are available. The loan application process is guided by an elaborate set of rules. Briefly, this process is as follows.

Required information for the loan processing are obtained from the customer in an application form. Additional information is gathered from sources mentioned in the application form. An applicant's mortgage amount cannot be greater than 2.5 times the total income. If the applicant has substantial outstanding personal loan, auto and credit card payments, then the mortgage amount cannot be greater than 1.5 times the total income. The applicant must have sufficient money in savings and/or checking accounts to pay for the down payment. In addition, the possession of a college degree is desirable because it indicates the applicant's reemployment potential if he/she is fired from the current job. The applicant should have had stable employment and income in the past 3 years. Also, the credit rating provided by a

credit reporting agency should be good. The loan evaluation process contains many additional rules, all of which cannot be stated in this article due to lack of space.

The bank has several loan officers with more than 10 years experience in loan evaluation. These officers know the loan evaluation process clearly and are willing to articulate the evaluation process. A knowledge engineer is available to acquire this knowledge and develop the system. The LES involves reasoning based on the information provided in the loan application. The reasoning process can be expressed in terms of rules and can be fitted into a production system scheme using IF....THEN rules. For example, some of the loan evaluation process rules can be stated as

IF

PERSONAL LOAN + AUTO + CREDIT CARD PAYMENTS > 20% TOTAL INCOME AND MORTGAGE AMOUNT > 1 .5 TOTAL INCOME

THEN

ACTION = DENY, CHANGE LOAN AMOUNT

ELSE

ACTION = CONTINUE;

IF

PERSONAL LOAN + AUTO + CREDIT CARD PAYMENTS < 20% TOTAL INCOME AND MORTGAGE AMOUNT > 2.5 TOTAL INCOME

THEN

ACTION = DENY, CHANGE LOAN AMOUNT

ELSE

ACTION = CONTINUE;

IF

COLLEGE DEGREE = YES AND NUMBER OF JOBS IN THE PAST 3 YEARS <= 2 AND INCOME SOURCES = VERIFIED CORRECT AND BREAK IN EMPLOYMENT= NO

THEN

JOB CONDITION = FAVORABLE;

.

.

(more rules)

.

.

IF

PAST PAYMENTS = TIMELY AND BOUNCED CHECK = NO AND CREDIT REPORT = GOOD

THEN

CREDIT RATING = FAVORABLE;

*

*

(more rules)

*

*

IF

ACTION = CONTINUE AND LIFE STYLE = FAVORABLE AND JOB CONDITION = FAVORABLE AND CREDIT RATING = FAVORABLE

THEN

LOAN = APPROVE

ELSE

LOAN = DENY;

In addition, the LES should explain why a loan was turned down because several states require banks to explain the reasons for loan denial as a fair credit reporting policy. A well structured and documented knowledge intensive problem such as this can be solved by an ES. The LES was easily developed using the ES software product, VP-EXPERT.

Alternatively, the LES can be developed using Artificial Neural Networks because there may be a correlation between the output (loan approved or loan not approved) and the set of inputs in the past loan applications, and that correlation may be useful to determine the output for new applications. Since the variables are known and a wide range of data on past loan applications are available, an ANN can be developed to implement the LES. The ANN development does not need the experts' involvement to identify rules and can be faster. An ANN was developed for this application using the software product NeuroShell. Fifty past loan applications data and their results were used to train the network. The ANN took about 9 hours to learn on a 286 micro computer running at 12.5 MHz clock speed and about 6 hours on 386 micro computer running at 20 MHz clock speed. When presented with new loan application data inputs, the trained ANN gave the following output: Loan Approved = 0.73 Loan Not Approved = 0.27

The system is 73 percent confident that the loan can be approved, adding some certainty to the decision. Presented with another loan application data, the ANN gave the following output: Loan Approved 0.45 Loan Not Approved 0.55.

In this example, neither output value is strong, calling for reevaluation of the loan by a human expert. This is a desirable feature of the system. The drawback is that the ANN cannot give an explanation for its decisions.

In terms of accuracy, both the ES and ANN perform well. Although both the ES and ANN can be used in this application, the requirement of the explanation system precludes the use of an ANN. An ES with its explanation system will be valuable to the bank in training new recruits. New loan officers can make use of the experienced loan officers' expertise in a simple way. In addition, when a change in the loan evaluation process is made, sample data may not be immediately available to train the ANN for future use. ESs need not wait for new samples and the rules can be changed in the ES for immediate use. Since the ANN can be developed in a short time and without the help of several experts, it is useful to validate the correctness of the ES.

CONCLUSIONS

The trend in information management has been to move from transaction processing systems to management information systems. Since management information systems provide only information, and a lot of it, decision support systems were developed to assist the manger in making quality decisions. The trend now is to move towards automated decisionmaking as facilitated by Expert Systems and Artificial Neural Networks. Rapid advances in technology will expedite the changeover.

REFERENEES

1. Murray, T. J. and Tanniru, M. R., "A Frame work for Selecting between Knowledge-based and Traditional Systems Design, Journal of MIS, Volume 4, Number 1, 1987, pp. 42-58.

2. Brownston, L. et. al, Programming Expert Systems in OPS5, Reading, MA: Addison-Wesley, 1985.

3. Goldberg, A. and Robson, D. Smalltalk-80; The Language and its Implementation, Reading, MA: Addison-Wesley, 1983.

4. Winston, P. H. Artificial Intelligence, Reading, MA: AddisonWesley, 1984.

5. Kowalski, R. Logic for Problem-Solving, New York, NY: North-Holland, 1979.

6. Yu, D. P. and Russel, R. L., "Neural Networks: Overview of Principles and Applications," Intelligent Systems Review, Volume 1, Number 2, 1988, pp. 29-47.

7. Wasserman, P.D. Neural Computing: Theory and Practice, New York, NY: Van Nostrand Reinhold, 1989.

8. Pigford, D. V., and Baur, G. Expert Systems for Business: Concepts and Applications, Boston MA: Boyd & Fraser. 7990.

9. NeuroShell: Reference Manual, Frederick, MD: Ward Systems Group, Inc., 1991

John R. Slater is an Expert Systems Analyst at Borders, Inc., Ann Arbor, MI.

Susan J. Hazen is a Programmer Analyst at ChoiceCare Inc., Cincinnati, OH.

Sachi Sakthivel, Ph.D. is Associate Professor of MIS at Bowling Green University, Bowling Green, OH.

GRAPHIC: Charts; Diagrams; References

JOURNAL-CODE: JSM

ABI-ACC-NO: 00771654

LOAD-DATE: August 1, 1995